www.icgst.com

Optimized Look-up-table for minimizing Critical Path in Distributed Arithmetic based FIR Filters

Sunita Badave[1], Anjali Bhalchandra[2]
[1] *Department of Electrical and Electronics Engineering, M.I.T., Aurangabad, 431028*
smb.eed@gmail.com.
[2] *Department of Electronics and Telecommunication Engineering, G.E.C., Aurangabad. 431005*
asbhalchandra@gmail.com

Abstract
Two new reduced critical path, look up table design techniques of distributed arithmetic (DA) based finite impulse response (FIR) filter are presented in this paper. Distributed arithmetic is one of the techniques, used to provide multiplier-free multiplication in the implementation of FIR filter. However it suffers from a sever limitation of exponential growth of look up table (LUT) with order of filter. Improved look-up-table optimization techniques are addressed here to design system architecture of FIR filter. In proposed technique, a single large LUT of conventional DA is replaced by number of smaller indexed LUT pages to restrict exponential growth and hence to reduce system access time. Selection module selects the desired value from desired LUT page and send for further filtering process. Trade off between access times of LUT pages and selection module helps to achieve minimum critical path so as to maximize the operating speed of filter. Further improvement in look-up-table design is achieved by reducing LUT data redundancy, which results into 40% rise operating frequency. Implementations are targeted to Xilinx ISE, Virtex IV devices with precision of 8 bit input samples. It is observed that, proposed designs perform significantly faster as compared to the conventional DA and existing DA based designs[1].

Keywords: *Critical Path, Multiplier less FIR filter, Distributed Arithmetic, LUT Design, Indexed LUT.*

Nomenclature

DA	Distributed Arithmetic
FIR	Finite Impulse Response
LUT	Look-up-table
DSP	Digital Signal Processing
CPCT	Critical Path Computation Time
LTI	Linear Time Invariant
NRE	Non-Recurring Engineering
ASIC	Application Specific Integrated Circuit
FPGA	Field Programmable Gate Arrays
OBC	Offset Binary Coding
LSB	Least Significant Bit
MSB	Most Significant Bit
ILUT	Indexed Look-up-table
DFG	Data Flow Graph
RILUT	Reduced Indexed Look-up-table3
MUX	Multiplexer
FDA	Filter Design Application
ISE	Integrated Software Environment

1. Introduction
Digital Signal Processing (DSP) systems are generally implemented using sequential circuits, where numbers of arithmetic modules in the longest path between any two storage elements are members of critical path. The Critical Path Computation Time (CPCT) determines the minimum feasible clock period and hence maximum allowable operating frequency of DSP system. Finite impulse response (FIR) digital filter is one of the widely used Linear Time Invariant (LTI) systems, has gained popularity in the field of digital signal processing due to its stability, linearity and ease of implementation. However, attention need to pay specifically while designing the high speed FIR filter, as CPCT is affected by both, system architecture as well as techniques used to design arithmetic modules. For such critical design of system architecture, fixed structure offered by Digital Signal Processor is not appropriate. However, high nonrecurring engineering (NRE) costs and long development time for application specific integrated circuits (ASICs) are making field programmable gate

[1] This study has been implemented at Government college of Engineering, affiliated to Dr. B.A.M.University, Aurangabad, Maharashtra, India.
The author is with the Department of Electrical and Electronics Engineering,M.I.T.,Aurangabad, Maharashtra,India. Her area of work is VLSI Signal processing.

arrays (FPGAs) more attractive for application specific DSP solutions. FPGA also offers design flexibility to arithmetic modules then ASICs.

For an N^{th} order FIR filter, each output sample is inner product of impulse response and input vector of latest N samples[1] given in equation (1).

$$Y(n) = \sum_{k=0}^{N-1} A_k X_{n-k} \quad (1)$$

For critical path minimization, direct implementation of equation (1) is not a cost effective solution because of two reasons. First critical path increases with the order of filter and second multiplier is an expensive arithmetic module with respect to area and computational time. More than two decade, many researchers [2-10] have worked on various multiplerless techniques for FIR filter design. In case of constant coefficient multiplication, LUT-multipliers [11-12,13] and distributed arithmetic [14-24] are two memory based approaches found in FIR filter design. Improved distributed arithmetic techniques are addressed here to design for system architecture for FIR filter. In recent years Distributed Arithmetic has gained substantial popularity due to its regular structure and high throughput capability, which results in cost-effective and efficient computing structure. This technique was first introduced by Croisier [14] and further development was carried out by Peled [15] for efficient implementation of digital filters in its serial form. Apart from its several advantages; DA based structure is facing a serious limitation of exponential growth of memory with order of filter. Many researchers [16-27] have addressed this problem, while dealing with this issues. Partial or full parallel structure with two and more than two bits [16,25] has been exploited to overcome the speed limitation, inherent to bit serial DA structure. Attempts were also been made to reduce memory requirement by recasting input data in Offset Binary Coding(OBC)[16], modified OBC and LUTless DA-OBC[19], instead of normal binary coding. Yoo *et al.* [22] extended this work and proposed a hardware efficient LUTless architecture, which gradually replaces LUT requirements with multiplexer/adder pairs. However gain in area reduction is achieved at the cost of increased critical path over the conventional design. LUT decomposition or slicing of LUT, proposed in [23], is one of the ways to restrict the exponential growth of memory. Though this technique has elucidated a problem of exponential growth of memory, involves the fact that latency and access time are the dependent parameters of level of decomposition.

As the operating speed of a filter is governed by worst case critical path, improved techniques are suggested in this paper to increase the speed of operation by reducing critical path. In proposed technique, a single large LUT of conventional DA is replaced by number of smaller indexed LUT pages to restrict exponential growth and to reduce system access time. Indexing of LUT pages eliminates the use of adders of existing techniques [16,17,19,22-24]. Selection module selects the desired value from desired page, and feed the value for further computation. Trade off between access times of LUTs and selection module helps to achieve minimum critical path so as to maximize the operating speed.

In organization of the paper, section 2 elaborates lookup table optimizations for conventional DA and proposed DA structures. Critical Path Computation Time analysis of previous and proposed techniques is given in section 3. Section 4 presents the realization of proposed architectures. In performance evaluation, component level access time analysis of proposed design is presented in section 5, followed by comparison of operating frequency of proposed and previous techniques. Paper is ended with conclusion, by section 6.

2. Look-up-table optimizations for DA algorithms

Distributed Arithmetic is one of the preferred methods of FIR filter implementation, as it eliminates the need for hardware multiplier. By this technique, sum-of-product terms in equation (1), can easily be transformed into addition. Assuming the multiplication with constant coefficients, let B be the word length of input samples, then, in unsigned binary form X(n) can be represented as:

$$X(n) = \sum_{i=0}^{B-1} x_{n,i} 2^i \quad (2)$$

where $x_{n,i}$ is the i^{th} bit of X(n). By Substituting the value of X(n) from equation (2) into equation (1), inner product can be expressed as:

$$Y(n) = \sum_{k=0}^{N-1} A_k \sum_{i=0}^{B-1} x_{ni} 2^i \quad (3)$$

Interchanging the sequence of summation in equation (3) results into:

$$Y(n) = \sum_{i=0}^{B-1} 2^i \sum_{k=0}^{N-1} A_k x_{ni} \quad (4)$$

Further, compressed form of equation (4), can be expressed as:

$$Y(n) = \sum_{i=0}^{B-1} 2^i \gamma \quad (5)$$

Where,

$$\gamma = A_0 x_{0,i} + A_1 x_{1,i} + \cdots + A_{N-2} x_{N-2,i} + A_{N-1} x_{N-1,i}$$

$X_{ni} \in \{0,1\}$

Thus equation (5) creates 2^N possible values of γ. All these values can therefore be precomputed and stored in form of look up table as shown in table 1. The filtering operation is performed by successively accumulating and shifting these precomputed values, based on the bit address formed by input samples, X(n).




LUT address bits				LUT contents
x_3	x_2	x_1	x_0	
0	0	0	0	0
0	0	0	1	A_0
0	0	1	0	A_1
0	0	1	1	$A_1 + A_0$
⋮	⋮	⋮	⋮	⋮
1	1	1	1	$A_3 + A_2 + A_1 + A_0$

Table 1. Conventional LUT design

2.1Proposed design of LUT-I: Indexed LUT: Execution time of conventionally designed LUT grows exponentially with the order of filter. A method is proposed to choose desired size of LUT for minimum Critical Path Computation Time(CPCT) of LUT unit. Let N= (n+m); where n and m are arbitrary positive integer, for the filter order N. A single large LUT size of 2^N, in conventional design is converted into 2^m LUT pages, each page with 2^n memory locations.

n- LUT address bits				LUT contents of each page
x_3	x_2	x_1	x_0	
0	0	0	0	I + 0
0	0	0	1	$I + A_0$
0	0	1	0	$I + A_1$
0	0	1	1	$I + A_1 + A_0$
0	1	0	0	$I + A_2$
0	1	0	1	$I + A_2 + A_0$
0	1	1	0	$I + A_2 + A_1$
0	1	1	1	$I + A_2 + A_1 + A0$
1	0	0	0	$I + A_3$
1	0	0	1	$I + A_3 + A_0$
1	0	1	0	$I + A_3 + A_1$
1	0	1	1	$I + A_3 + A_1 + A_0$
1	1	0	0	$I + A_3 + A_2$
1	1	0	1	$I + A_3 + A_2 + A_0$
1	1	1	0	$I + A_3 + A_2 + A_1$
1	1	1	1	$I + A_3 + A_2 + A_1 + A_0$

Table 2. Proposed LUT design

Page number	m - Address Bits		Index terms *I* for LUT pages
	x_5	x_4	
0	0	0	0
1	0	1	A_4
2	1	0	A_5
3	1	1	$A_5 + A_4$

Table 3. Indexed term for each LUT page

Applying this concept to the equation (5), number of terms in γ can be divided into two groups: n LSB terms and m MSB terms. It is represented by:

$$\gamma = \left(A_0 x_{0,i} + A_1 x_{1,i} + \cdots + A_{n-2} x_{n-2,i} + A_{n-1} x_{n-1,i}\right) + \left(A_n x_{n,i} + \cdots + A_{n+m-1} x_{n+m-1,i}\right) \quad (6)$$

LSB n bits, defines the size of each LUT page, however, MSB m bits defines number of LUT pages. Instead of consisting coefficient sum in conventional look up table, LUT of proposed design consists of indexed-sum-of-filter-coefficients. A page selector module selects desired output from one of the LUT pages, addressed by m bits. A desired combination of n and m facilitates to select the minimum execution time of LUT page and page selector module, to attain maximum operating frequency.

LUT page structure of 6th order filter, for n=4 and m=2 and indexed term of each page, is elaborated in table 2 and table 3 respectively. Each LUT page contains summation of filter coefficients and index term I.

It can be also proved by Shannon's expansion:

$$\begin{aligned} & f(A_{n+m-1}, A_{n+m-2}, \ldots, A_1, A_0) \\ &= \overline{A_{n+m-1}}\,\overline{A_{n+m-2}} f(0,0, A_{n-1}, \ldots, A_1, A_0) \\ &+ \overline{A_{n+m-1}}\, A_{n+m-2} f(0,1, A_{n-1}, \ldots, A_1, A_0) \\ &+ A_{n+m-1}\,\overline{A_{n+m-2}} f(1,0, A_{n-1}, \ldots, A_1, A_0) \\ &+ A_{n+m-1}, A_{n+m-2} f(1,1, A_{n-1}, \ldots, A_1, A_0) \end{aligned} \quad (7)$$

Applying it to LUT page structure of 6th order filter, for n=4 and m=2, with splitting terms A5 and A4, modeling LUT will be:

$$\begin{aligned} & LUT(A_5, A_4, A_3, A_2, A_1, A_0) \\ &= \overline{A_5}\,\overline{A_4}(evenpage\ 0) + \overline{A_5} A_4 (oddpage\ 1) \\ &+ A_5\,\overline{A_4}(evenpage\ 2) + A_5 A_4 (oddpage\ 3) \end{aligned} \quad (8)$$

Equation.8 can be realized as shown in figure 1.

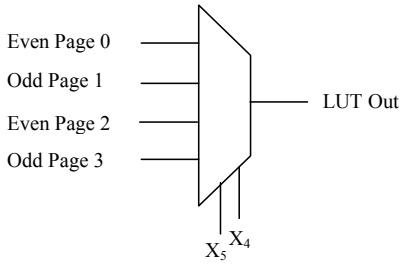


Figure 1. Realization of equation (8)

2.2 Proposed design of LUT-II: Reduced Indexed LUT(ILUT): Speed gain achieved in ILUT DA technique is at the cost of area overheads when compare with previous DA structures[21,22]. Efforts have further extended to reduce the area by reducing memory requirement. A new area efficient scheme is proposed, which need only half of the memory to that of indexed LUT DA technique.

Design proposed in section 2.1 consists of 2^m memory pages. It can be reduced 2^{m-1} memory pages, by reducing data redundancy between two consecutive pages. Referring table 3, contents of even numbered pages (page 0 and page 2) are exactly same as that of the odd numbered pages (page 1 and page 3) respectively, except the 'term of difference' A_4.

If this 'term of difference' is added to the contents of even page of LUT depending on the status of x_i, it helps to eliminate all odd pages. Structure modified by this technique, reduces the number of LUT pages half to that of indexed LUT structure. As LUT access

time is a major contributor of CPCT, reduction in it leads to reduction in overall CPCT.

3. Critical path computation time analysis of proposed architecture

In this section, CPCT analysis of conventional DA, LUTless DA [23,26], sliced DA [20,21,27,28]and proposed DA based FIR filter techniques are elaborated. Designs proposed in [26, 27] are taken into consideration, as these designs are found more comparable with the proposed designs.

3.1 Conventional DA based FIR filter: Conventional form of distributed arithmetic FIR filter given in figure 2, consisting of bank of input registers, look-up-table (LUT) unit, and accumulator/shifter unit.

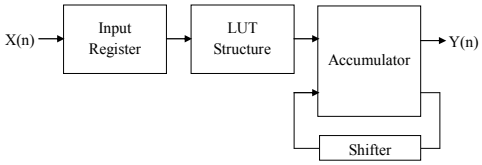


Figure 2. Functional Block Diagram of Conventional DA based FIR Filter

Serially arriving input data values X(n) are stored in parallel form, in register bank. During each clock cycle, registers take a right shift and output a word, which is used to address the LUT. Shift and accumulate for successive B cycles generates Y(n).

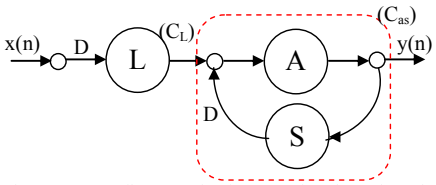


Figure 3. Data flow graph of conventional DA based FIR filter

Data flow graph(DFG) of conventional DA based FIR filter, as shown in figure 3. It consists of nodes L, A and S, represents LUT, accumulator and shifter respectively. As feedback path is consisting of a delay, critical path is defined by L and A. Access times of these nodes are C_L and C_{as} respectively. Access Time of node L increases exponentially with order of filter, however, that of node A is almost independent. Thus CPCT of conventional DA based FIR filter can be expressed as:

$$CPCT_{(cnv)} = C_L + C_{as} \quad (9)$$

3.2 LUTless DA based FIR filter: Elimination of LUT is an attempt found in [23,26], to overcome exponential growth of LUT. In such LUTless structure (figure 4.), LUT is successively replaced by multiplexer-adder pair. On-line data generated by multiplexers are added to generate the output.

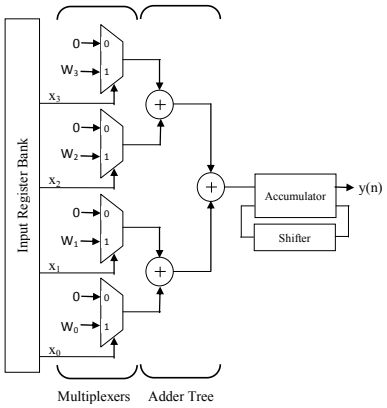


Figure 4. LUTless DA based FIR filter

DFG of LUTless DA based FIR filter, shown in figure 5, consists of multiplexer node M, adder nodes T_a and accumulator node A. Though the number of multiplexers is governed by order of filter, access time of only one multiplexer contributes in CPCT, as they are operating concurrently.

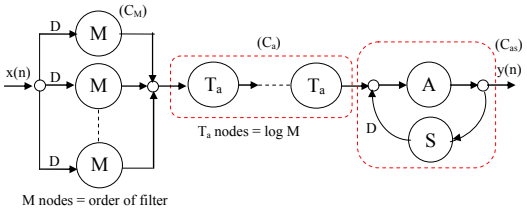


Figure 5. Data flow graph of LUTless DA based FIR filter

Assuming the adders are arranged in 4:2 form in adder tree, $\log_2(N)$ adders are taken into consideration while calculating its access time C_a. It will be expressed as:

$$C_a = \log_2 N \times T_a \quad (10)$$

Thus C_a is highly found to be filter order dependent as indicated by equation (10). Critical path computational time(CPCT) of structure becomes:

$$CPCT_{(LUTless)} = C_M + C_a + C_{as} \quad (11)$$

where C_M - access time of multiplexer.

C_{as} – access time of accumulator/shifter unit.

3.3 Sliced LUT DA based FIR filter: Another well-known attempt found in[20,21,27,28] to restrict the exponential growth of LUT, is the use of multiple memory banks. Latest, Longa *et al.*[27], highlighted that, FIR filter structure will be an area efficient structure if a single large LUT is replaced by number of 4-input, smaller LUTs. However, this arrangement leads to put a burden of adders, which are required to add partial terms generated by each smaller LUT sections. Generally such LUT arrangement is referred as partitioning of LUT or slicing of LUT. Architectural details of sliced DA based FIR filter is shown in figure 6.

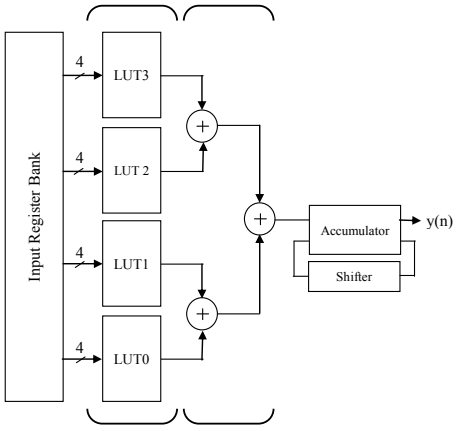


Figure 6. Sliced LUT DA based FIR filter

Data flow graph(DFG) of sliced LUT DA based FIR filter, shown in figure 7, consists of concurrently operating LUT nodes L_s, adder nodes Ta, accumulator A and shifter node S. In this type of architecture, number of adders in adder tree is governed by number of slices. Assuming the order of filter is divisible by 4, for N^{th} order FIR filter, N/4 will be number of slices and (N/4)-1 will be the number of adders. Thus LUT node L_s, $[\log_2(N/4)]$ adders and accumulator are the members of critical path. So the CPCT of the structure will be:

$$CPCT_{(Slice)} = (C_{SL} + C_a + C_{as}) \quad (12)$$

Where

C_{SL} = access time of one slice of LUT.
C_a = access time of adder tree
= $[\log_2(N/4)]T_a$
T_a = access time of an adder.
C_{as} = access time of accumulator/shifter

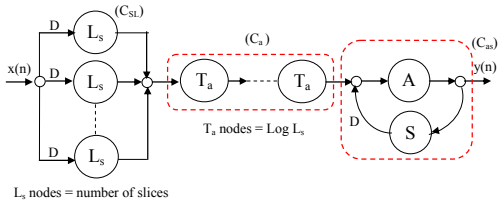


Figure 7. Data flow graph of sliced LUT DA based FIR filter

Access time of LUT get reduced from C_L to C_{SL} due to slicing technique, however it has added the over heads of adder tree access time C_a in $CPCT_{(slice)}$.

3.4 Indexed LUT DA based FIR filter: Exponential growth of LUT with order of filter is playing a key role not only in hardware complexity but also in implementation of optimum CPCT. Earlier solutions offered has restricted the exponential growth of LUT[26,27], however it has increased the burden of access time of adder tree. So an attempt is made, to eliminate the use of single large LUT as well as adders. In proposed design of Indexed LUT(ILUT) DA based filter structure, node L of conventional design is replaced by smaller, desirably indexed LUTs L_i and multiplexer M. Data flow graph(DFG) of the proposed design derived from equation (6), is shown in figure 8. CPCT of this structure, contributed by L_i-M-A nodes, will now be:

$$CPCT_{(Index)} = C_i + C_m + C_{as} \quad (13)$$

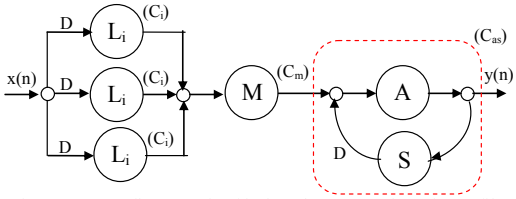


Figure 8. Data flow graph of indexed LUT DA based FIR filter

Access time C_i and C_m are interdependent. The trade off of an exponentially varying LUT with linearly varying multiplexer size helps to choose optimum CPCT of a structure. Hence, improves overall operating frequency of filter. It also eliminates the need of adder tree, which further helps to improve the operating frequency.

3.5 Reduced Indexed LUT(RILUT) DA based FIR filter: CPCT is greatly affected complexity of filter structure. Further work is extended to reduce memory space of ILUT by reducing LUT data redundancy. This reduced indexed LUT(RILUT) design reduces, number of LUT pages to half, than the number of LUT pages in indexed LUT design. This causes to reduce the size of multiplexer M, and hence its access time C_{mr} in DFG. Though the number of LUT pages reduced to half, there will no change in C_i, as all the pages operated concurrently. DFG for this RILUT DA structure is similar to DFG of ILUT DA with modified multiplier access time C_{mr}. CPCT of this structure, contributed by L_i-M-A nodes, will now be:

$$CPCT_{(rindex)} = C_i + C_{mr} + C_{as} \quad (14)$$

4. Realization of proposed architecture

Realization of LUT optimizations for DA based FIR filters are elaborated in this section. These are built up with three major components; bank of input shift registers, look-up-table unit and accumulator /shifter unit. Apart from these units, it needs a control circuit to control filtering operation.

4.1 Input register bank: Register Bank is built up with N number of serial-in parallel-out shift register. It accepts X(n), N input samples serially and arrange them in parallel form. For every clock pulse, register contents takes right shift and output a word. Number of right shifts is governed by precision of input word.

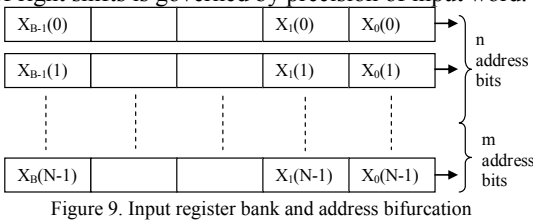


Figure 9. Input register bank and address bifurcation

Address generated by register bank, shown in figure 9, is split into two address groups *n* and *m*, where *n* and *m* are arbitrary integer. LSB *n* bits define address of LUT, whereas number of LUT pages is defined by *m* bits.

4.2 LUT optimization technique-I: Indexed LUT(ILUT): Indexed LUT DA based FIR filter comprises with indexed LUT pages, each of size 2^n and a multiplexer unit as a page selection module as shown in figure 10. All LUT pages are parallel - addressed by LSB n bits. Each page consists of indexed precomputed filter coefficient sum. A desired output term of LUT pages is selected by multiplexer, addressed by MSB m bits. For an example, structural details of 6th order FIR filter, with n=4 and m=2, is shown in figure 10. Four LUT pages, each with 16 locations are connected in parallel. A multiplexer unit of size 4:1 selects an appropriate output for further stage.

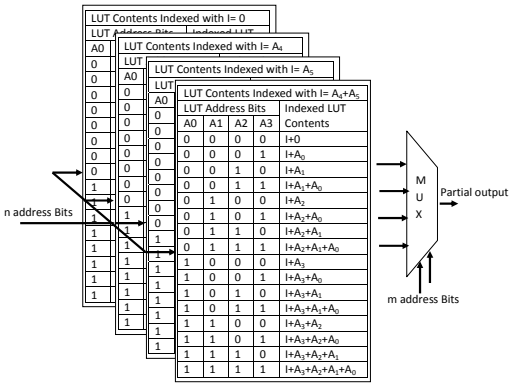


Figure 10.Proposed structure of indexed LUT unit

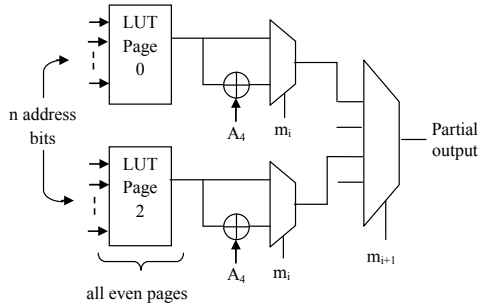


Figure 11. Proposed structure of reduced indexed LUT unit

4.3 LUT optimization technique-II: Reduced Indexed LUT(RILUT): Data redundancy of ILUT structure is removed in optimization technique-II, RILUT DA based FIR filter, by an external glue logic in combinational form. Figure 11 shows, even and odd page distribution of LUT pages of 6th order FIR filter. Indexed terms of page0 and page1 are 0 and A_4 respectively. Similarly, page2 and page3 are A5 and A4+A5 respectively. Contents of page0 and page2 are same as that of page1 and page3, if A_4 term is added in contents of page 0 and page 2. Thus modified LUT unit comprises only with even numbered by adding A_4 externally.

4.4 Accumulator and Shifter Unit: Accumulator and shifter are two separate combinational units, however jointly these are responsible for calculating the dot product term of filter output. Its hardware complexity is greatly influenced by the way LUT addressed and shift given by accumulator/shifter unit. Instead of giving left shift to the terms generated by LUT, it is more convenient to give right shift to the accumulator contents; it reduces the hardware complexity of structure.

4.5 Control unit: It is a finite state machine, shown in figure12, defines sequence of operation and has overall control on filtering operation.

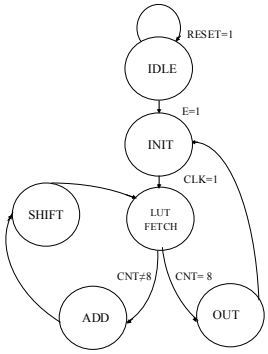


Figure 12. Control unit of proposed structure

Filtering operation remains in idle state with application of reset. It starts with enable signal E and takes iteration equal to input precision for every clock cycle. At the end of count it gives filter output and operation begins with next fetch cycle.

Order of Filter	Address Line distribution		LUT Unit Access time analysis		Operating freq. in MHz		Area in terms of gate count	
	n	m	C_i in ns	C_m in ns	Proposed Design-I	Proposed Design-II	Proposed Design-I	Proposed Design-II
8	7	1	6.58	3.6	151.389	210.011	1847	1486
	6	2	5.45	4.06	160.937	208.810	1912	1766
	5	3	5.02	4.46	155.876	216.224	1923	1680
	4	4	4.65	4.8	184.834	225.251	2050	1745
	3	5	4.65	5.16	169.544	238.875	2252	1829
	2	6	4.6	5.5	168.714	240.252	2779	2090
	1	7	3.84	6.1	176.625	214.381	2129	1735
7	6	1	5.45	3.6	189.92	227.167	1567	1232
	5	2	5.02	4.06	180.874	230.104	1600	1304
	4	3	4.65	4.46	183.441	255.678	1533	1250
	3	4	4.65	4.8	191.18	254.113	1593	1386
	2	5	4.6	5.16	182.45	260.94	1912	1500
	1	6	3.84	5.5	190.13	234.446	1501	1225
6	5	1	5.02	3.6	190.3	255.552	1222	1015
	4	2	4.65	4.06	192.417	277.507	1206	1101
	3	3	4.65	4.46	205.495	277.507	1235	1077
	2	4	4.6	4.8	190.389	277.288	1431	1138
	1	5	3.84	5.16	192	254.874	1226	1337
5	4	1	4.65	3.6	206.793	279.731	990	881
	3	2	4.65	4.06	228.645	279.731	969	875
	2	3	4.6	4.46	239.664	278.275	1093	874
	1	4	3.84	4.8	215.736	273.052	987	878
4	3	1	4.65	3.6	242.93	387.93	833	723
	2	2	4.6	4.06	242.93	387.271	833	757
	1	3	3.84	4.46	244.09	387.271	827	766

Table 4. Access time analysis of LUT unit modules

5. Performance evaluation

Each node of proposed structure is critically analyzed for the range of filter from 4 to 8, which further can be extended to desired order of filter. For a particular order of filter N, filter performance is evaluated for all possible combinations of n and m.

As computation time of accumulator/shifter node C_{as} is almost independent with the order of filter, access time C_i and C_m are the only contributors for CPCT of proposed structure. Variations in C_i , C_m , operating frequency and area in terms of gate count of proposed structures for each combination of m and n are given in table 4. Graphical representation for 8th order FIR filer is shown in figure 13. It indicates that, access time of LUT page C_i increases exponentially with n, at the same time access time of multiplexer C_m decreases linearly with it.

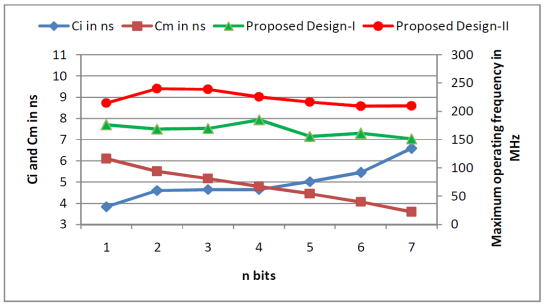


Figure 13. Relation between access time analysis of LUT unit modules and operating frequency of 8th order FIR filter

If f_{max} is maximum operating frequency and T_{sample} is the minimum time required to process each output sample, then

$$T_{sample} \geq CPCT$$
$$\geq C_i + C_m + C_{as} \quad (15)$$

As

$$f_{max} = 1/ T_{sample}$$
$$f_{max} \leq 1/ C_i + C_m + C_{as} \quad (16)$$

As CPCT minima of filter is obtained at the point of intersection of LUT access time and MUX access time, which leads to maximum operating frequency. Thus filter design corresponds to these values of m and n will be treated as optimized design. Performance is further getting improved in proposed design-II and proved to be high speed, area efficient.

Results obtained by the proposed techniques are compared with Conventional DA, LUTless DA and Sliced LUT DA Techniques. Previously presented techniques[22,23] were implemented on Altera Stratix FPGA chip. To surmount the platform differences, these techniques are faithfully implemented on same platform as that of the proposed techniques. A comparative study of maximum operating speed of existing and proposed DA based filter techniques is presented in table 5.

Order of filter	Operating frequency of DA based filter in MHz				
	Conventional DA	LUT less DA	Sliced DA	Proposed DA design-I	Proposed DA design-II
4	242.4	242.93	240.13	244.09	387.271
5	239.01	239.06	220.03	239.664	278.27
6	200.95	174.07	200.12	205.495	277.50
7	184.65	175.50	185.68	191.18	260.94
8	176.22	174.28	167.72	184.834	240.252

Table 5. Operating frequency comparison of various architectures

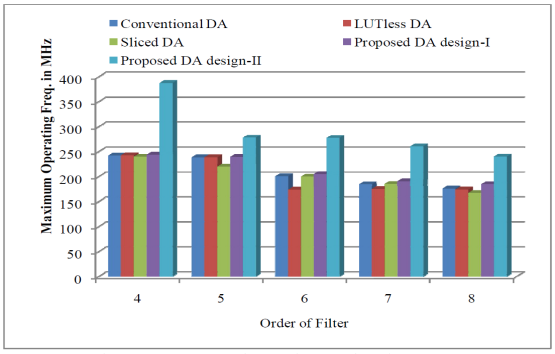


Figure 14. Comparison of operating frequency

Variations in maximum operating frequency of these techniques with order of filter are shown in figure 14. It indicates one of the obvious observations is that operating frequency reduces with the order of filter.

Order of filter	Area in terms of gate count				
	Conventional DA	LUT less DA	Sliced DA	Proposed DA design-I	Proposed DA design-II
4	841	833	841	833	723
5	984	978	981	990	881
6	1208	1165	1121	1222	1015
7	1599	1331	1214	1567	1232
8	1888	1514	1546	1847	1486

Table 6. Area comparison of various architectures

It is also observed that operating frequency of proposed techniques-I is higher than conventional DA and previously presented DA[22,23] techniques round about 5%. It further gets improved by 40% with proposed technique-II. It has also observed that proposed technique-II has saved 25% of area than earlier techniques.

Desired filter coefficients are obtained from FDATool, a special toolbox of MATlab, which are truncated and scaled to 8-bit precision. Xilinx Integrated Software Environment (ISE) is used for performing synthesis and implementation of the designs.

All the designs are performed on vertex xc2vp FPGA and synthesized for maximum performance. To validate the correct functionality, each implementation is simulated with the simulation tool provided by Xilinx. Tests are carried out using random inputs. As an example, simulation test results of 6th order FIR filter, designed with ILUT-DA based technique is shown in figure 14.

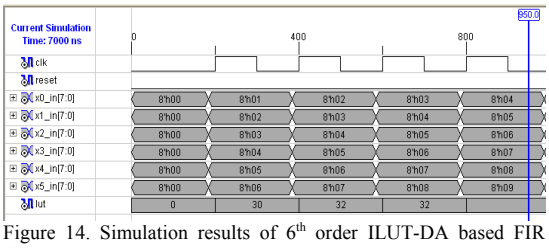


Figure 14. Simulation results of 6th order ILUT-DA based FIR filter

Structural complexities given in table 6, of N^{th} order filter are analyzed and performances are compared for random input samples x(n). Word length of input sample and filter coefficient is assumed to be of B bits, which makes size of input register bank to be same for all designs under consideration. Latency and throughput found same in all DA based structures; however operating speed of individual technique makes the value to differ.

For implementation of N^{th} order conventional DA based FIR filter requires memory array of 2^N x B bits and the size of decoder is $N:2^N$. CPCT of the structure is $(C_L + C_{as})$, increases exponentially due to exponential rise in C_L, however C_{as} is independent with order of filter. Thus it is almost constant in all structures. Structural complexities of conventional DA based FIR filters are considered as bench marks for performance comparison.

Slicing of single large memory reduces the memory requirement of design from 2^N X B of conventional DA to (a X 2^l) X B; where a and l are the factors of N. Thus decoder also get changed from single N: 2^N to a, $l:2^l$. As multiple terms are generated by this technique, need at least a-1 adders to generate coefficient sum as partial term. A single large LUT is replaced by smaller LUTs, reduces LUT access time from C_L to C_{SL}, however it adds adder access time C_a, tending to increase CPCT of structure.

LUTless technique selects filter coefficient on-line by multiplexer, eliminates the need of memory and corresponding decoder at the cost of N-1 adders. As LUT is replaced by multiplexers and adders, C_M and C_a are the contributors of CPCT, which are highly filter order dependent.

In proposed technique, indexing of LUT pages reduces its access time C_i instead of C_L as well as eliminates C_a as a prime contributor of CPCT of LUTless and sliced LUT DA based techniques. It adds a small burden of LUT page selection module C_m, to CPCT of structure. However it leads to reduce overall CPCT, leading to increase in operating frequency. LUT complexity has further reduced in LUT optimization technique-II due modified C_{mr}.

6. Conclusion

For high speed FIR filter implementation in distributed arithmetic, the exponential rise of memory access time with the filter coefficients has always been considered to be a fundamental drawback. Innovative techniques to reduce CPCT of FIR filter is designed and implemented successfully, this has leads to increase in operating frequency by 5% when filter designed with LUT optimization technique-I and about 40% with design of LUT optimization technique-II. It has also proved that optimization technique-II to be an area economical technique as it has offered area gain of 25%, which was not attainable in proposed technique-I.

Proposed designs have restricted exponential growth of memory as well as need of adders, which was mandatory in earlier presented designs. Proposed designs have proved faster than previous work

.

Order of filter	Structural Complexities				
	Conventional DA	LUTless DA	Sliced DA	Proposed DA design-I	Proposed DA design-II
Input Register	NxB	NxB	NxB	NxB	NxB
Memory Bits	$M_C= 2^N$x B	-	$M_S= (a \times 2^l) \times B$	$M_I= (2^m \times 2^n)$ xB	$M_r= (2^{m-1} \times 2^n)$ xB
Decoder	N: 2^N	-	$a(l:2^l)$	2^m $(n:2^n)$	2^{m-1} $(n:2^n)$
Number of Adders	-	N-1	a-1	-	-
Depth of Adders	-	$B+\log_2 N$	$B+\log_2 a$	-	-
Multiplexers	-	-	-	2^m:1	2^{m-1}:1
CPCT	$C_L + C_{as}$	$C_M+ C_a+ C_{as}$	$C_{SL} +C_a+ C_{as}$	$C_i+C_m+C_{as}$	$C_i+C_{mr}+C_{as}$
Latency	B+1	B+1	B+1	B+1	B+1
Throughput	B+2	B+2	B+2	B+2	B+2

Table 6. Structural complexity of previous and proposed designs

7. References

[1] Mitra S. K. Digital Signal Processing-A Computer Based Approach. Third Edition, India.Tata McGraw Hill, 2008.

[2] Samueli Henry. An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients. IEEE Transactions on Circuits and Systems. 36(7), pp.1044-1047, 1989.

[3] Evans Joseph B. An efficient FIR filter architectures suitable for FPGA implementation. Proc. IEEE International Symposium on Circuits and Systems (ISCAS), pp.226-228, 1993.

[4] Oh Woo Jin, Lee Yong Hoon. Implementation of programmable multiplierless FIR filters with powers-of-two coefficients. IEEE Transactions on Circuits and Systems-II Analog and Digital Signal Processing, 42(8), pp.553-556, 1995.

[5] Khoo Kei-Yong, Kwentus Alan, and Alan. Willson. (1996) A programmable FIR digital filter using CSD coefficients, IEEE Journal of Solid-State Circuits, 2(6), pp.869-874,1996.

[6] Dawoud D. S. Realization of pipelined multiplier - free FIR digital filter. Proc. IEEE Africon Conference, pp.335–338,1999.

[7] Kosunen Marko, Halonen Halonen Kari. A programmable FIR filter using serial-in-time multiplication and canonic signed digit coefficients. Proc. . 7th IEEE International Conference on Electronics, Circuits and Systems(ICECS), pp.563-566, 2000.

[8] Tan Kah-Howe, Leong Wen Fung, Kaluri Kadambari, Soderstrand M A. and Johnson Louis G. FIR filter design program that matches specifications rather than filter coefficients results in large savings in FPGA resources. Proc. IEEE International Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, 2, pp.1349-1352, 2001.

[9] Tang Zhangwen, Zhang Jie and Min Hao. A High-speed, programmable, CSD coefficient FIR filter, IEEE Transactions on Consumer Electronics, 48(4), pp.834-837, 2002.

[10] Yeung K. S. and Chan S. C. Multiplier-less digital filters using programmable sum-of-power-of-two (SOPOT) coefficients. Proc. IEEE International Conference on Field-Programmable Technology (FPT), pp.78-84, 2002.

[11] Meher Pramod Kumar. New look-up-table optimizations for memory-based multiplication. Proc.12th IEEE International Symposium on Integrated Circuits (ISIC), pp.663-666, 2009.

[12] Meher Pramod Kumar. New Approach to Look-Up-Table Design and Memory-Based Realization of FIR Digital Filter. IEEE Transactions on Circuits and Systems—I,57(3), pp.512-603, 2010.

[13] Parhi Keshab K. VLSI Digital Signal Processing- Design and Implementation. India John Wiley & Sons,1999.

[14] Croisier A., Esteban D. J., Levilion M. E., and Rizo V. Digital filter for PCM encoded signals U.S. Patent3 777 130, 1973.

[15] Peled A. and Liu B. A new hardware realization of digital filters, IEEE Trans. Acoust. Speech, Signal Process, 22(6), pp.456–462, 1974.

[16] White S.A. Applications of the distributed arithmetic to digital signal processing: A Tutorial Review. IEEE ASSP Mag,6(3), pp.5–19, 1989.

[17] Burleson Wayne P., Scharf Louis L. VLSI design of inner-product computers using distributed arithmetic, Proc. of the IEEE International Symposium on Circuits and Systems(ISCAS),pp.158-161, 1989.

[18] Babic Rudi, Solar Mitja, Stiglic Bruno. High order FIR digital filter realization in distributed arithmetic. Proc. Mediterranean Electrotechnical Conference, pp.367-370, 1991.

[19] Choi Jung-Pal, Shin Seung- Cheol, Chung Jin- Gyun. Efficient ROM size reduction for distributed arithmetic. IEEE International Symposium on Circuits and Systems(ISCAS), 2, pp.61-64, 2000.

[20] Chang T. S. and Jen C.W. Hardware-efficient pipelined programmable FIR filter design. Proc. IEEE on Computers and Digital Techniques,148(6), pp.227-232, 2001.

[21] Chen Chin-Chao, Lin Tay-Jyi, Liu Chih-Wei, and Jen Chein-Wei. Complexity aware design of DA-based FIR filters. Proc. IEEE Asia-Pacific Conference on Circuits and Systems, pp.445-448, 2004.

[22] Yoo Heejong and Anderson David V. Hardware-efficient distributed arithmetic architecture for high-order digital filters. Proc.IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),5, pp.125-128, 2005.

[23] Longa Patrick and Miri Ali (2006) Area-efficient FIR filter design on FPGA using distributed arithmetic, IEEE International Symposium on Signal Processing and Information Technology, pp.248-252, 2006.

[24] Mehendale M., Sherlekar S. D., and Venkatesh G. Area-delay Tradeoff in distributed arithmetic based implementation of FIR filters, Proc. 10th International Conference of VLSI Design, pp.124-129, 1997.





[25] Jeng Shiann-Shiun, Lin Hsing-Chen, and Chang Shu-Ming FPGA implementation of FIR filter using M-bit parallel distributed arithmetic, IEEE International Symposium on Circuits and Systems (ISCAS), pp.875-878, 2006.

[26] Meher Pramod Kumar, Chandrasekaran Shrutisagar, Amira Abbes. FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic. IEEE Transactions on Signal Processing, 56(7), pp.3009-3017, 2008.

[27] B.K.Mohanty,P.K.Meher. A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications. Transactions on Very Large Scale Integration (VLSI) Systems, issue 99,1-9, 2015.

Biographies



Sunita Mukund Badave received the B.E. degree in Electrical (Electronics Specialization) from Shivaji University in 1989 and M.E.Degree in Electrical from Dr. B. A. M. University, Aurangabad, India, in 1998. She is currently working toward the Ph.D. degree in Electronics at Dr. B. A. M. University. Her research interests include architectures and circuit design for digital signal processing. Mrs. S.M. Badave is *Member* of the Institute of Electronics and Telecommunication Engineers (IETE), life Member of Institute of Engineers (IE) and life member of Indian Society for Technical Education(ISTE)India. She is also member of International Association of Engineers (IAENG).



Anjali S. Bhalchandra received the B.E. Electronics and Telecommunication degree and M.E. Electronics degree in 1985 and 1992 respectively. She has completed her Ph.D. in Electronics from S.R.M.University, Nanded, India, in 2004. She has a scientific and technical background covering the areas of Electronics and Communication. Currently, she is Professor of Electronics and Telecommunication Engineering Department in Government College of Engineering, Aurangabad. Her research interest includes image processing, signal processing and communication. She has published more than 50 technical papers in various reputed journals and conference proceedings. Dr. Bhalchandra is a Fellow of the Institution of Engineers (IE), India and life member of Indian Society for Technical Education(ISTE) India.